The Smart Party: A Personalized Location-aware Multimedia Experience

Kevin Eustice, V. Ramakrishna, Nam Nguyen, Peter Reiher
UCLA Computer Science Department
University of California, Los Angeles
Los Angeles, California
{kfe,vrama,songuku,reiher}@cs.ucla.edu

***Abstract*— We describe the Smart Party, a new ubiquitous computing application for the home environment. This application gathers musical preferences for guests in different rooms of a party in a user's house. Based on their preferences and available media, it chooses a music play list for each room, adjusting to changing membership as guests move through the party. We describe the application, its architecture, and our implementation of it, including key performance characteristics.**

Keywords-ubiquitous computing, social media, configuration management

I. INTRODUCTION

The personal audio player has become ubiquitous, and is now attached to the hips, pockets, and hearts of hundreds of millions of users. Users exchange digital audio content with each other socially, sharing their digital audio players in a mostly ad hoc manner [1]. New digital audio players, such as the Microsoft Zune, are designed to allow users to share music wirelessly with others, but in practice connecting and configuring the devices has proven troublesome for many users [2,3]. This is an exciting technology, but further development and streamlining is necessary to captivate the consumer.

The Smart Party represents the next generation in social media sharing: it allows user devices to seamlessly and transparently detect, configure, and interact socially in physical locations. At the Smart Party, users bring their media preferences and media library on their portable device. Their devices are automatically configured for their environment, and interact to select the media that will play in the various rooms in which the party is situated. Per-room play lists are dynamically generated based on the current user population.

There are many other exciting possibilities that this model of interaction opens up. Within a Smart Party, user data and context—preferences, media, metadata, location, social memberships, application interests, etc.—could be used collectively and collaboratively to not only select audio and video, but also alter environmental settings, coordinate power management, and identify interested participants for group activities.

These kinds of applications are exciting, but existing networking infrastructure is not equipped to handle this type of organization and device management. To enable these types of applications, networking infrastructure must be able to deal with the realities of many embedded devices needing to connect, interact and coordinate with one another—ideally in an extensible manner, and with minimal user intervention.

We have developed a supporting middleware for pervasive computing called *Panoply* that addresses these and other related issues. Panoply enables simple creation, configuration, and discovery of computational contexts that support communication-based groups, location-based groups, and interest- and task-based groups. These groups, called *spheres of influence*, organize related peers, and scope communication and configuration.

Section 2 describes our vision, design, and implementation of the Smart Party. Section 3 discusses our experiences with a deployed Smart Party, and also provides basic performance measurements. Section 4 presents related work and Section 5 contains our conclusions.

II. THE SMART PARTY

In our vision of the Smart Party, a group of people attend a gathering hosted at someone's home. Each person carries a small mobile device that stores its owner's music preferences and song collection. The party environment consists of a series of rooms, each equipped with speakers. The home is covered by one or more wireless access points that have been secured; additionally, neighboring homes may possess access points that are also visible from the party space. This starting setup is shown in Figure 1, which depicts our group of guests who have just arrived at the Smart Party environment.

As each guest arrives, his mobile device securely and automatically associates with the correct network to connect it to the Smart Party infrastructure. As party attendees move within the party environment, each room programs an audio playlist based on the communal music preferences of the current room occupants, and the content they have brought to the party. Guests automatically and dynamically collaborate with the host network, which manages their collective preferences and steers the music choices. One vision of a fully formed and configured Smart Party is shown in Figure 2. For example, Rock Music tends to predominate in the Family Room because Bob, Mary and Charlie are in that room and share a common preference for that genre.

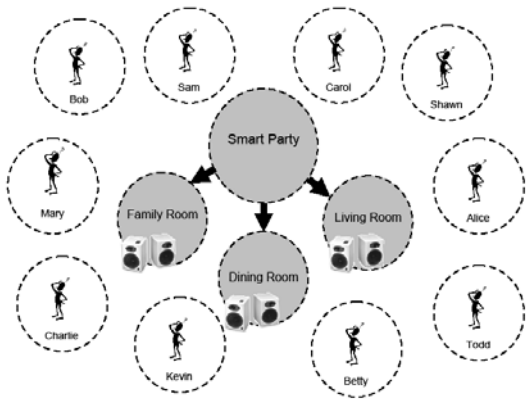

Figure 1. Prior to Smart Party Configuration

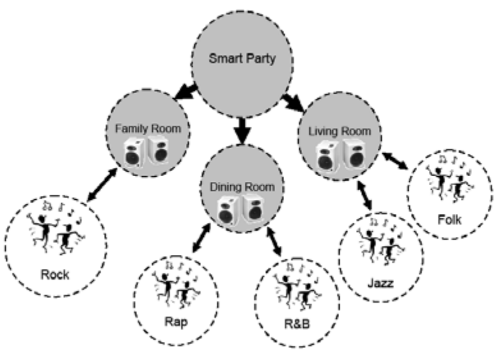

Figure 2. A Fully Configured Smart Party

This application features secure device configuration and connection, flexible localization map dissemination, and mediated interaction of ad hoc proximity-based social groups. The Smart Party application maps to other applications involving group interactions and configuration and connectivity based on environments.

A. Technical Challenges

The Smart Party is a location- and group-aware application. There are a number of key functions that were required to realize this application, including: auto-configuration of groups of devices, group awareness, localization, automatic association with the correct network, detection of movement and group changes, and also dynamic management of playlists and media selection.

At a high-level, these functional requirements can be broken down into the following categories: *Configuring and Authorizing Guest Devices, Localizing Guest Devices, Media Suggestion and Search.* Each of these functions is implemented as a set of services.

B. Design and Implementation

The functions of context-awareness, dynamic configuration, and management are not unique to our Smart Party scenario, but are shared by many similar applications. We have implemented these and other common services in Panoply, an active middleware for managing devices and device applications[4].

1) Panoply and Spheres of Influence

The core representational unit of Panoply is the *Sphere of Influence*, which can represent an individual device or a group of devices united by a common interest or attribute such as physical location, application, or social relationship. Spheres unify disparate notions of "groups", such as device clusters and social networks, by providing a common interface and a standard set of discovery and management primitives.

Panoply provides group management primitives that allow the creation and maintenance of spheres of influence, including discovery, joining, and cluster management. A publish/subscribe event model is used for intra- and inter-sphere communication. Events are propagated between devices and applications, subject to scoping constraints embedded in events and interest. Every sphere scopes policy and contains a policy manager [5] that monitors the environment, mediates interactions and negotiates agreements.

Panoply supports the design of applications that express interests and communicate through events. Panoply applications (e.g., the Smart Party) can create custom events, and designate the scope and destination of such events.

2) The Smart Party as a Panoply Application

Smart Party attendees arrive at the hosting location with their portable devices, each of which supports its own device sphere. The party environment is represented by a social sphere, and each room within the party has its own location sphere.

The Smart Party contains three sub-applications, (1) A master application that runs in the Smart Party social sphere, (2) A location application running in each location sphere, and (3) A user device application running in each user device sphere

The attendees' device spheres are pre-configured to enable them to merge seamlessly with the party social sphere. Before the party, an invitation program creates and delivers party invitations in the form of *vouchers* that contain a MAC address as well as other necessary wireless configuration data. Typically, the MAC address encoded in the voucher corresponds to a wireless access point within the actual party environment. Invitations are disseminated to users via e-mail or through a direct sphere connection, if one is formed after invitation creation. A voucher is a generic multipurpose cryptographic credential similar to a Simple Public Key Infrastructure (SPKI) certificate.

When a user device sphere observes the MAC address described in the voucher, it activates the network configuration encoded in that voucher to connect to the appropriate wireless network. Next, it launches the user application and connects to the Smart Party, providing its voucher to the Smart Party sphere to gain entry. The Smart Party master provides a *semantic localization map* to the user device sphere, allowing it to navigate the party. Details of Panoply localization support are beyond the scope of this paper, but are available in [6].

When the user application detects a change in semantic location (i.e., the user moves between rooms), it issues a *join* *event* to its own sphere, requesting that the device join the location sphere specified by the map. Once connected, the application receives a *membership event* and severs the connection to any previous room through a *leave* event. The Sphere join process, and also the production and verification of vouchers, are carried out through automated negotiation protocols managed by the respective spheres' policy managers.

When there is at least one active member in a location sphere, the sphere's location application executes a three-stage media provisioning protocol to determine the local group's musical preference, identify a source for the audio content, and retrieve the content. The first stage, or suggestion stage, consists of a user-typed search event that is locally scoped to the location sphere and its current members. Each member can vote for a set of songs or genres. The location application acts as a coordinator, evaluating the search responses and selecting a song based on common interest or common genre. Once a song is selected, the location application sends out a search event to all members at the party. If a member responds that he possesses the content, he will receive a delivery request event and start sending the content. When the audio content is retrieved, it is delivered to a set of speakers attached to the location.

Security Issues: Security and access control policies can be specified for every Panoply sphere and enforced by the sphere's policy manager. A Panoply-based smart party must allow only authorized guests to participate. It makes sure of this by specifying a policy that requires every prospective participant to produce a valid guest voucher as authorization through negotiation [5]. Once a party is instantiated, secure TLS channels are used for all intra-party communication. Additionally, our smart party design allows dynamic manipulation of the playlists by authorized party hosts. If an ordinary guest attempted to manipulate the playlist, a negotiation would be initiated. Unless the guest could produce a valid voucher that indicated host privileges, playlist access would be denied. Lastly, digital rights issues will certainly arise whenever audio files are shared across computers. This issue is somewhat orthogonal to our Smart Party design and prototype. Panoply policy could be used to restrict the transfer and playback of controlled material, given appropriate digital rights management (DRM) support. Similarly, assuming appropriate metadata is available, mature material could be similarly restricted if minors are present at the Smart Party.

III. Results and Experiences

We have deployed the Smart Party within an office suite at UCLA, and it has been in operation for over a year. Our experimental setup is illustrated in Figure 3, which consists of a map that represents our research lab, and is drawn to scale. Three cubicles within our lab (A, D and E) were selected to represent rooms in a smart party, and localization maps were built accordingly. The computer on which the sphere hosting the master application runs was physically located in cubicle D, and maintains logical connections to every mobile device in the lab. For simplicity, all three of our spheres hosting the location applications run on the above machine as well. Each party location consists of an ASUS WL-500g Deluxe wireless router that can be programmed using the OpenWrt firmware. Speakers are attached to each router, and play audio as dictated by a script running on the associated router. Mobile devices, including IBM Thinkpad T42 laptops and Sony Vaio UX280 uPCs, associate with the router in the vicinity, and transfer audio files to the latter when required. Currently, devices do not prefer a specific router, but Panoply can easily manage router assignment based on network load or user and device location.

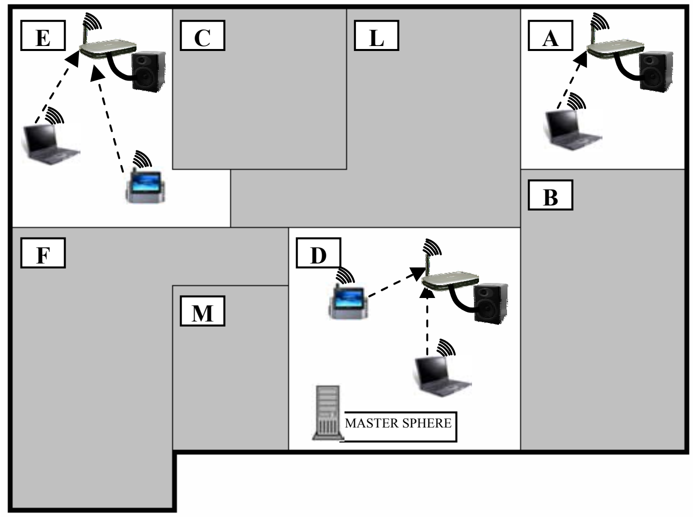

Figure 3. Experimental Setup at the Laboratory for Advanced Systems Research (LASR), Computer Science Department, UCLA

Our prototype relies on a simple and efficient networking framework based on 802.11-based wireless MAC protocols for network association and DHCP for address acquisition. As described in Section II.B, pre-deployed vouchers are used by every mobile device to select the appropriate wireless network to associate with, thereby initiating interaction with the master sphere. Once localized to a cubicle, a similar network association and address acquisition procedure is carried out with the wireless access point that represents that cubicle.

Figure 4. Sample GUI Presented to Smart Party Participants

A number of students have experienced and experimented with the Smart Party; additionally, it has also been demonstrated to numerous visitors to our lab. To augment the user experience, and enable more involvement and control, we designed a Panoply application that presented each user with a GUI that indicates what music is currently playing in each cubicle. It also presents the user's music collection and preferences, and not only allows him/her to manipulate such preferences, but also to dynamically modify the playlist. A sample GUI is illustrated in Figure 4 below. A typical policy associated with the location spheres prevents playlist manipulation by anyone other than party hosts, who can prove their authorization by producing a valid voucher through automated negotiation.

We have taken basic performance measurements of the various component phases of the Smart Party—these are summarized in Table I below. The figures in the right-hand column indicate the recorded mean overheads for all phases of the Smart Party, from the moment a user device detects the Smart Party environment through to participation in the Smart Party media provisioning protocol. Means were calculated over data gathered from 1,320 runs of the Smart Party. Results are reported with 99% confidence intervals.

TABLE I. OVERHEAD FOR PHASES OF SMART PARTY INITIALIZATION.

Smart Party Phase	Mean Time
Match Voucher	1373 ± 86 ms
Configure Network Settings	334 ± 3 ms
Acquire DHCP Address	8000 ± 146 ms
Launch Application & Connect to Party Sphere	2797 ± 38 ms
Negotiate for Party Access	1149 ± 18 ms
Receive Loc. Map	852 ± 45 ms
Apply Loc. Map	1959 ± 12 ms
Localize	16441 ± 817 ms
Begin Party Participation	1776 ± 21 ms
Total Elapsed Time	34788 ± 827 ms

After the user device detects the Smart Party environment, it takes approximately 14 seconds on average to configure itself and join the principal Smart Party social sphere. Another 20 seconds, the bulk of which is due to localization, are required to join the appropriate location-specific sphere and fully participate in the Smart Party. These operations are completely transparent to the user. When we consider that simply entering the environment and greeting the host may by itself take 30 seconds to several minutes, the configuration overhead is quite reasonable. Further refinement of the localization technology, research that is orthogonal to the aims of this project, could result in significantly reduced overhead.

Our experiences with the Smart Party have led to several observations.

Localization is an imperfect technology. The 802.11-based localization scheme we use in our prototype sometimes generates false negatives and positives. A transient failure in localization often resulted in the user device leaving the location sphere—if the user was alone, the music would fade out, even if the user quickly rejoined the sphere. When a user really did move to a new room, there might be gaps in the audio playback during the transition. Therefore, we modified the user device application so the device leaves a location only if the user has joined a new location sphere or has left the party entirely. This caused smoother transitions between rooms, and reduced transient audio glitches, suggesting that location-driven applications must be designed to be tolerant of localization failures. Sanity-checking technologies, such as transit time calculations, or accelerometer measurements, where available, could also act as double-check mechanisms.

We were able to build and deploy a basic implementation of the Smart Party in approximately 1-2 weeks using Panoply. Two more weeks were spent incorporating features like localization map dissemination, mechanisms for creation and dissemination of vouchers, and building a voucher manager and database for every sphere. Other location and social group-oriented applications could also be built and deployed in a similar timeframe. The only work involved would be in design, and determining the kinds of events sent and received by such an application.

The Panoply precursor to the Smart Party was an Interactive Narrative [6], which was set on the UCLA campus and involved participants playing the roles of characters within a story. The participants would wander across campus with their mobile devices, which would deliver content appropriate to their location and history. Participants could perform virtual actions that would affect the progress and eventual outcome of the narrative. As in a Smart Party scenario, the participants comprised a social group, and could join and leave location spheres. Content delivery and action updates were done through events and mediated by policy. Localization maps and vouchers were used for similar purposes as in the Party, though in a less automated fashion.

These two representative applications built and deployed with little effort demonstrate our middleware's versatility. Of course, we cannot anticipate all possible features required to support all possible applications, and pushing too much functionality into the middleware would be undesirable for performance reasons. We anticipate this to be a learning process over the course of the next few years, and an application like the Smart Party is a promising start in that direction.

IV. RELATED WORKS

Projects like MusicFX [7], Adaptive Radio [8] and FlyTrap[9] explored social music experiences that adapted to user preferences. They focus on algorithms for equitable music selection in a single group in a single shared environment. Our Smart Party application supports a more general notion of groups and allows interesting interactions, such as sharing music preferences and media.

Jukola [10] is an interactive MP3 Jukebox device that allows a group of people in a public space to choose the music being played via a democratic process. Members in each group share a single handheld device and confer orally to find a common vote. This process is totally transparent to the users in the case of Smart Party.

Several works have been focused on the visual interface [11,12] and collaborative preference elicitation [13] to help members in a group to reach a consensus. These techniques can be applied to the Smart Party in the future to improve its usability and user-friendliness.

Network-based social group awareness augments human senses by alerting a user to the presence of nearby social group members. Early research in this area demonstrated the need for devices that alert group members of proximity to each other [14]. Dodgeball [15] is an SMS-based system that allows friends to share their location via a centralized service, which then alerts them to nearby friends. Reno [16], built on Place Lab [17] supports *social disclosure* of location. Friends and family members can choose to automatically disclose their location, real or fabricated, to social peers via a cell-phone-based service. Wang [18] investigated social management for pervasive computing applications using a model of social group interactions and a framework that addresses some of the issues involved in social group management. In contrast, the Smart Party application used the semantic localization map provided by the Panoply framework to discover relevant groups. This mechanism provides a general view of group formations and interactions and allows applications to join and discover relevant groups flexibly.

V. CONCLUSION

We are beginning to see the advantages of dynamically grouping highly mobile consumer devices when they congregate in one location. Just as individual use of personal computers matured into dynamic group experiences across the Internet, mobile consumer devices are destined to cooperate to provide their users with better social experiences, bringing people together, rather than keeping them apart. Since the users of mobile devices are often in the same physical space, these experiences can complement real human contact, rather than substitute for it, a frequent criticism of Internet communities. The Smart Party application is an example of how to take advantage of the mobility and ubiquity of consumer computing resources to provide new and exciting user experiences.

The Smart Party application points out several interesting possibilities: shared media among multiple users, leveraging the preferences and desires of individual users to improve overall experiences, and the use of groups to organize and manage collaborative applications. The Panoply middleware framework is a good example of the kind of software support required to help further development of these applications.

One important lesson of our experiences with the Smart Party is that the mobile ubiquitous environment is complex and unpredictable. Tools that allow simplifying generalizations (like Panoply's formalization of groups) will simplify application development. Better debugging and evaluation tools are also required for these environments to deal with the frequently complex and unpredictable behaviors of large numbers of mobile devices cooperating and competing.

VI. REFERENCES

[1] L. Kahney. "Feel Free to Jack Into My IPod." Wired Magazine. November 21, 2003.

[2] S. Dredge. "How Zune Music Sharing Should Work" Techdigest.tv June 28, 2007. http://techdigest.tv/2007/06/how_zune_music.html

[3] S. Jayson. "Microsoft's Xbotch" http://www.fool.com/investing/value/2007/06/07/microsofts-xbotch.aspx.

[4] K. Eustice, L. Kleinrock, S. Markstrum, G. Popek, V. Ramakrishna, and P. Reiher . "Enabling Secure Ubiquitous Interactions." 1st International Workshop on Middleware for Pervasive and Ad-Hoc Computing (IWMPAC2003).

[5] V. Ramakrishna, K. Eustice, and P. Reiher, "Negotiating Agreements Using Policies in Ubiquitous Computing Scenarios," Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07).

[6] K. Eustice, V. Ramakrishna, A. Walker, M. Schnaider, N. Nguyen and P. Reiher, "nan0sphere: Location-Driven Fiction for Groups of Users," Proceedings of the 12th International Conference on Human-Computer Interaction (HCII 2007).

[7] J. F. McCarthy and T. D. Anagost, MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, ACM Press, New York, 1998.

[8] D. Chao, J. Balthrop, and S. Forrest, Adaptive Radio: Achieving Consensus Using Negative Preferences, Proceedings of GROUP '05.

[9] A. Crossen, J. Budzik, and K. Hammond. Flytrap: Intelligent Group Music Recommendation. Proceedings of the 7th international conference on Intelligent user interfaces. San Francisco, California, USA, 2002

[10] K. O'Hara, M. Lipson, M. Jansen, A. Unger, H. Jeffries, P. Macer. Jukola: Democratic Music Choice in a Public Space. Proceedings of the 2004 Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques.

[11] A. Jameson. More Than the Sum of its Members: Challenges for Group Recommender Systems. Proceedings of the 2004 Working Conference on Advanced Visual Interfaces.

[12] A. Jameson, S. Baldes, and T. Kleinbauer. Two Methods for Enhancing Mutual Awareness in a group Recommender System. Proceedings of the 2004 Working Conference on Advanced Visual Interfaces.

[13] C. Plua and A. Jameson. Collaborative Preference Elicitation in a Group Travel Recommender System. Proceedings of the AH 2002 Workshop on Recommendation and Personalization in eCommerce.

[14] L. E. Holmquist, J. Falk, and J. Wigstrm, Supporting Group Collaboration with Inter-Personal Awareness Devices, Journal of Personal Technologies, 3 (1-2), Springer Verlag, 1999.

[15] Dodgeball.com—Mobile Social Software. http://www.dodgeball.com/

[16] I. Smith, S. Consolvo, J. Hightower, J. Hughes, G. Iachello, and A. LaMarca, Social Disclosure Of Place: From Location Technology to Communication Practice, Proceedings of Pervasive 2005, Munich, Germany, May 2005.

[17] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, Place Lab: Device Positioning Using Radio Beacons in the Wild, Proceedings of Pervasive 2005, Munich, Germany, May 2005.

[18] B. Wang, J. Bodily, S. K. S. Gupta, Supporting Persistent Social Groups in Ubiquitous Computing Environments Using Context-Aware Ephemeral Group Service, Proceedings of PerCom 2004, Orlando, Florida, March 2004.